

Amdt. dated March 21, 2005
Reply to Office action of 12/21/2004

Serial No. 09/755,814
Docket No. TUC920000052US1
Firm No. 0018.0084

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1. (Original) A computer implemented method for selecting a code image during a reboot routine, comprising:

maintaining multiple code images in a memory device;
executing a first operation routine;
incrementing a first counter if the first operation routine succeeds;
executing a second operation routine;
incrementing a second counter if the second operation routine succeeds; and
using the first and second counters to select one of the code images from the memory device to execute.

2. (Original) The method of claim 1, further comprising:

designating one code image as non-operational if the first counter is a first value and the second counter is a second value, wherein one other code image not designated as non-operational is selected to execute.

3. (Original) The method of claim 2, wherein the first value is greater than zero and the second value is zero.

4. (Original) The method of claim 2, further comprising:

receiving an update to the code image;
determining whether one code image is designated as non-operational; and

Amdt. dated March 21, 2005
Reply to Office action of 12/21/2004

Serial No. 09/755,814
Docket No. TUC920000052US1
Firm No. 0018.0084

overwriting the code image designated as non-operational with the received update to the code image if one code image is designated as non-operational.

5. (Original) The method of claim 4, further comprising:
determining an earliest version of the code images in the memory device; and
overwriting the determined earliest version of the code image if one code image is not designated as non-operational.

6. (Original) The method of claim 4, further comprising:
determining whether one code image is corrupted; and
if one code image is corrupted, overwriting the corrupted code image with the received update before determining whether one code is non-operational.

7. (Original) The method of claim 1, wherein the first operation routine comprises a reboot routine and the second operation routine comprises an initialization routine.

8. (Original) The method of claim 7, further comprising:
incrementing the second counter if the initialization routine successfully completed;
rebooting if the initialization routine failed; and
performing another iteration of all previous steps after rebooting.

9. (Original) The method of claim 7, further comprising:
selecting one copy of the code image, wherein the executed initialization routine is a component of the selected code image, wherein the selected code image is designated as non-operational if the first counter is the first value and the second counter is the second value;
and

Amdt. dated March 21, 2005
Reply to Office action of 12/21/2004

Serial No. 09/755,814
Docket No. TUC920000052US1
Firm No. 0018.0084

selecting one other copy of the code image if the selected code image is designated as non-operational.

10. (Original) The method of claim 1, wherein the code image comprise different versions of the code image.

11. (Previously presented) A computer implemented method for selecting a code image during a reboot, comprising:

maintaining multiple code images in a memory device, wherein the code images include a function routine to perform an operation after initialization;

executing a reboot routine and incrementing a first counter if the reboot routine succeeds;

executing an initialization routine and incrementing a second counter if the second operation routine succeeds;

executing the function routine in one code image and incrementing a third counter associated with the one code image including the executed function routine if the function routine succeeds; and

using the first, second, and third counters to select one of the multiple code images from the memory device to execute.

12. (Original) The method of claim 11, further comprising:

designating one code image as operational if the first, second, and third counters satisfy at least one threshold value, wherein the code image designated as operational is automatically selected from the memory device to execute after subsequent reboot operations.

13. (Original) The method of claim 11, further comprising:

Am dt. dated March 21, 2005
Reply to Office action of 12/21/2004

Serial No. 09/755,814
Docket No. TUC920000052US1
Firm No. 0018.0084

designating one code image as non-operational if the first, second, and third counters satisfy at least one threshold value, wherein one other code image not designated as non-operational is selected from the memory device and executed.

14. (Previously presented) The method of claim 13, wherein the threshold value for the third counter is zero and wherein the at least one threshold value for the first and second counters is greater than zero.

15. (Original) The method of claim 13, further comprising:
incrementing the second counter if the initialization routine successfully completed;
rebooting if the initialization or function routine failed; and
performing another iteration of all previous steps after rebooting.

16. (Original) The method of claim 11, wherein the code image includes multiple function routines, wherein there is one counter for each of the multiple function routines, and further comprising:

designating one code image as operational if the first counter, second counter, and each counter associated with a function routine satisfy at least one threshold value, wherein the code image designated as operational is automatically selected from the memory device to execute after subsequent reboot operations.

17. (Previously presented) The method of claim 1, wherein one operation routine comprises a function routine to perform a device specific operation.

18. (Previously presented) A computer system for selecting a code image during a reboot routine, comprising:

Amdt. dated March 21, 2005
Reply to Office action of 12/21/2004

Serial No. 09/755,814
Docket No. TUC920000052US1
Firm No. 0018.0084

a processor;
a memory device maintaining multiple code images, wherein the processor is capable of accessing the memory device;
program logic executed by the processor, wherein the program logic causes the processor to perform:
(i) executing a first operation routine;
(ii) incrementing a first counter if the first operation routine succeeds;
executing a second operation routine;
(iii) incrementing a second counter if the second operation routine succeeds; and
(iv) using the first and second counters to select one of the code images from the memory device to execute.

19. (Original) The system of claim 18, wherein the program logic is further capable of causing the processor to perform:

designating one code image as non-operational if the first counter is a first value and the second counter is a second value, wherein one other code image not designated as non-operational is selected to execute.

20. (Original) The system of claim 19, wherein the first value is greater than zero and the second value is zero.

21. (Original) The system of claim 19, wherein the program logic is further capable of causing the processor to perform:

receiving an update to the code image;
determining whether one code image is designated as non-operational; and

Amtd. dated March 21, 2005
Reply to Office action of 12/21/2004

Serial No. 09/755,814
Docket No. TUC920000052US1
Firm No. 0018.0084

overwriting the code image designated as non-operational with the received update to the code image if one code image is designated as non-operational.

22. (Original) The system of claim 21, wherein the program logic is further capable of causing the processor to perform:

 determining an earliest version of the code images in the memory device; and
 overwriting the determined earliest version of the code image if one code image is not designated as non-operational.

23. (Original) The system of claim 21, wherein the program logic is further capable of causing the processor to perform:

 determining whether one code image is corrupted; and
 if one code image is corrupted, overwriting the corrupted code image with the received update before determining whether one code is non-operational.

24. (Original) The system of claim 18, wherein the first operation routine comprises a reboot routine and the second operation routine comprises an initialization routine.

25. (Original) The system of claim 24, wherein the program logic is further capable of causing the processor to perform:

 incrementing the second counter if the initialization routine successfully completed;
 rebooting if the initialization routine failed; and
 performing another iteration of all previous steps after rebooting.

26. (Original) The system of claim 24, wherein the program logic is further capable of causing the processor to perform:

Amdt. dated March 21, 2005
Reply to Office action of 12/21/2004

Serial No. 09/755,814
Docket No. TUC920000052US1
Firm No. 0018.0084

selecting one copy of the code image, wherein the executed initialization routine is a component of the selected code image, wherein the selected code image is designated as non-operational if the first counter is the first value and the second counter is the second value; and

selecting one other copy of the code image if the selected code image is designated as non-operational.

27. (Original) The system of claim 18, wherein the code image comprise different versions of the code image.

28. (Previously presented) A computer system for selecting a code image during a reboot, comprising:

a processor;

a memory device maintaining multiple code images, wherein the processor is capable of accessing the memory device, and wherein the code images include a function routine to perform an operation after initialization;

program logic executed by the processor, wherein the program logic causes the processor to perform:

(i) executing a reboot routine and incrementing a first counter if the reboot routine succeeds;

(ii) executing an initialization routine and incrementing a second counter if the initialization routine succeeds; and

(iii) executing the function routine in one code image and incrementing a third counter associated with the code image including the executed function routine if the function routine succeeds; and

Amtd. dated March 21, 2005
Reply to Office action of 12/21/2004

Serial No. 09/755,814
Docket No. TUC920000052US1
Firm No. 0018.0084

(iv) using the first, second and third counters to select one of the code images from the memory device to execute.

29. (Original) The system of claim 28, wherein the program logic is further capable of causing the processor to perform:

designating one code image as operational if the first, second, and third counters satisfy at least one threshold value, wherein the code image designated as operational is automatically selected from the memory device to execute after subsequent reboot operations.

30. (Original) The system of claim 28, wherein the program logic is further capable of causing the processor to perform:

designating one code image as non-operational if the first, second, and third counters satisfy at least one threshold value, wherein one other code image not designated as non-operational is selected from the memory device and executed.

31. (Previously presented) The system of claim 30, wherein the threshold value for the third counter is zero and wherein the at least one threshold value for the first and second counters is greater than zero.

32. (Original) The system of claim 30, wherein the program logic is further capable of causing the processor to perform:

incrementing the second counter if the initialization routine successfully completed; rebooting if the initialization or function routine failed; and performing another iteration of all previous steps after rebooting.

Amdt. dated March 21, 2005
Reply to Office action of 12/21/2004

Serial No. 09/755,814
Docket No. TUC920000052US1
Firm No. 0018.0084

33. (Original) The system of claim 28, wherein the code image includes multiple function routines, wherein there is one counter for each of the multiple function routines, and further comprising:

designating one code image as operational if the first counter, second counter, and each counter associated with a function routine satisfy at least one threshold value, wherein the code image designated as operational is automatically selected from the memory device to execute after subsequent reboot operations.

34. (Previously presented) The system of claim 18, wherein one operation routine comprises a function routine to perform a device specific operation.

35. (Original) An article of manufacture for selecting a code image during a reboot routine, wherein the article of manufacture includes code in a computer readable medium capable of causing a processor to perform:

maintaining multiple code images;
executing a first operation routine;
incrementing a first counter if the first operation routine succeeds;
executing a second operation routine;
incrementing a second counter if the second operation routine succeeds; and
using the first and second counters to select one of the code images to execute.

36. (Original) The article of manufacture of claim 35, wherein the article of manufacture code is further capable of causing the processor to perform:

designating one code image as non-operational if the first counter is a first value and the second counter is a second value, wherein one other code image not designated as non-operational is selected to execute.

Amdt. dated March 21, 2005
Reply to Office action of 12/21/2004

Serial No. 09/755,814
Docket No. TUC920000052US1
Firm No. 0018.0084

37. (Original) The article of manufacture of claim 36, wherein the first value is greater than zero and the second value is zero.

38. (Original) The article of manufacture of claim 36, wherein the article of manufacture code is further capable of causing the processor to perform:

receiving an update to the code image;

determining whether one code image is designated as non-operational; and

overwriting the code image designated as non-operational with the received update to the code image if one code image is designated as non-operational.

39. (Original) The article of manufacture of claim 38, wherein the article of manufacture code is further capable of causing the processor to perform:

determining an earliest version of the code images; and

overwriting the determined earliest version of the code image if one code image is not designated as non-operational.

40. (Original) The article of manufacture of claim 38, wherein the article of manufacture code is further capable of causing the processor to perform:

determining whether one code image is corrupted; and

if one code image is corrupted, overwriting the corrupted code image with the received update before determining whether one code is non-operational.

41. (Original) The article of manufacture of claim 35, wherein the first operation routine comprises a reboot routine and the second operation routine comprises an initialization routine.

Amtd. dated March 21, 2005
Reply to Office action of 12/21/2004

Serial No. 09/755,814
Docket No. TUC920000052US1
Firm No. 0018.0084

42. (Original) The article of manufacture of claim 41, wherein the article of manufacture code is further capable of causing the processor to perform:

incrementing the second counter if the initialization routine successfully completed;
rebooting if the initialization routine failed; and
performing another iteration of all previous steps after rebooting.

43. (Original) The article of manufacture of claim 41, wherein the article of manufacture code is further capable of causing the processor to perform:

selecting one copy of the code image, wherein the executed initialization routine is a component of the selected code image, wherein the selected code image is designated as non-operational if the first counter is the first value and the second counter is the second value; and

selecting one other copy of the code image if the selected code image is designated as non-operational.

44. (Original) The article of manufacture of claim 35, wherein the code image comprise different versions of the code image.

45. (Previously presented) An article of manufacture for selecting a code image during a reboot, wherein the article of manufacture includes code in a computer readable medium capable of causing a processor to perform:

maintaining multiple code images, wherein the code images include a function routine to perform an operation after initialization;

executing a reboot routine and incrementing a first counter if the reboot routine succeeds;

executing an initialization routine and incrementing a second counter if the initialization routine succeeds;

Arndt, dated March 21, 2005
Reply to Office action of 12/21/2004

Serial No. 09/755,814
Docket No. TUC920000052US1
Firm No. 0018.0084

executing the function routine in one code image and incrementing a third counter associated with the code image including the executed function routine if the function routine succeeds; and

using the first, second and third counters to select one of the code images to execute.

46. (Original) The article of manufacture of claim 45, wherein the article of manufacture code is further capable of causing the processor to perform:

designating one code image as operational if the first, second, and third counters satisfy at least one threshold value, wherein the code image designated as operational is automatically selected to execute after subsequent reboot operations.

47. (Original) The article of manufacture of claim 45, wherein the article of manufacture code is further capable of causing the processor to perform:

designating one code image as non-operational if the first, second, and third counters satisfy at least one threshold value, wherein one other code image not designated as non-operational is selected and executed.

48. (Previously presented) The article of manufacture of claim 47, wherein the threshold value for the third counter is zero and wherein the at least one threshold value for the first and second counters is greater than zero.

49. (Original) The article of manufacture of claim 47, wherein the article of manufacture code is further capable of causing the processor to perform:

incrementing the second counter if the initialization routine successfully completed; rebooting if the initialization or function routine failed; and performing another iteration of all previous steps after rebooting.

Amdt. dated March 21, 2005
Reply to Office action of 12/21/2004

Serial No. 09/755,814
Docket No. TUC920000052US1
Firm No. 0018.0084

50. (Original) The article of manufacture of claim 45, wherein the code image includes multiple function routines, wherein there is one counter for each of the multiple function routines, wherein the article of manufacture code is further capable of causing the processor to perform:

designating one code image as operational if the first counter, second counter, and each counter associated with a function routine satisfy at least one threshold value, wherein the code image designated as operational is automatically selected to execute after subsequent reboot operations.

51. (Previously presented) The article of manufacture of claim 35, wherein one operation routine comprises a function routine to perform a device specific operation.

52. (Previously presented) The method of claim 1, wherein executing the first and second operation routines, and incrementing the first and second counters are performed during a reboot operation, and wherein the code images include implementations of the first and second counters.

53. (Previously presented) The method of claim 52, wherein logic for executing the first and second operation routines, logic for incrementing the first and second counters, logic for using the first and second counters to select one of the code images, and the code images are implemented in firmware that can be updated.

54. (Previously presented) The method of claim 7, wherein the initialization routine is included in the code image and is capable of initializing variables and parameters used by the code image.

Amdt. dated March 21, 2005
Reply to Office action of 12/21/2004

Serial No. 09/755,814
Docket No. TUC920000052US1
Firm No. 0018.0084

55. (Previously presented) The method of claim 11, wherein the code images include implementations of the first, second, and third counters, and wherein the code images include implementations of the reboot routine, the initialization routine, and the function routine.

56. (Previously presented) The system of claim 18, wherein executing the first and second operation routines, and incrementing the first and second counters are performed during a reboot operation, and wherein the code images include implementations of the first and second counters.

57. (Previously presented) The system of claim 56, wherein logic for executing the first and second operation routines, logic for incrementing the first and second counters, logic for using the first and second counters to select one of the code images, and the code images are implemented in firmware that can be updated.

58. (Previously presented) The system of claim 24, wherein the initialization routine is included in the code image and is capable of initializing variables and parameters used by the code image.

59. (Previously presented) The system of claim 28, wherein the code images include implementations of the first, second, and third counters, and wherein the code images include implementations of the reboot routine, the initialization routine, and the function routine.

60. (Previously presented) The article of manufacture of claim 35, wherein executing the first and second operation routines, and incrementing the first and second counters are performed during a reboot operation, and wherein the code images include implementations of the first and second counters.

Arndt, dated March 21, 2005
Reply to Office action of 12/21/2004

Serial No. 09/755,814
Docket No. TUC920000052US1
Firm No. 0018.0084

61. (Previously presented) The article of manufacture of claim 60, wherein logic for executing the first and second operation routines, logic for incrementing the first and second counters, logic for using the first and second counters to select one of the code images, and the code images are implemented in firmware that can be updated.

62. (Previously presented) The article of manufacture of claim 41, wherein the initialization routine is included in the code image and is capable of initializing variables and parameters used by the code image.

63. (Previously presented) The article of manufacture of claim 45, wherein the code images include implementations of the first, second, and third counters, and wherein the code images include implementations of the reboot routine, the initialization routine, and the function routine.

64. (Previously presented) The method of claim 1, wherein the second operation routine is an initialization routine that performs an initialization of the code image by initializing variables and parameters used by the code image.

65. (Previously presented) The system of claim 18, wherein the second operation routine is an initialization routine that performs an initialization of the code image by initializing variables and parameters used by the code image.

66. (Previously presented) The article of manufacture of claim 35, wherein the second operation routine is an initialization routine that performs an initialization of the code image by initializing variables and parameters used by the code image.

Amtd. dated March 21, 2005
Reply to Office action of 12/21/2004

Serial No. 09/755,814
Docket No. TUC920000052US1
Firm No. 0018.0084

67. (Previously presented) The method of claim 1, wherein the code images include a function routine to perform an operation after initialization, and wherein the function routine is also used to select one of the code images.

68. (Previously presented) The system of claim 18, wherein the code images include a function routine to perform an operation after initialization, and wherein the function routine is also used to select one of the code images.

69. (Previously presented) The article of manufacture of claim 35, wherein the code images include a function routine to perform an operation after initialization, and wherein the function routine is also used to select one of the code images.